ISSUES CONCERNING MATERIAL REMOVAL SHAPE ELEMENT VOLUMES (MRSEVs)

Thomas R. Kramer

Guest Researcher, NIST Robot Systems Division &

Research Associate, Catholic University

March 26, 1992

NISTIR 4804

Abstract

In machining discrete parts, Material Removal Shape Element Volumes (MRSEVs) may be used to convey shape information from process planning to NC-programming. This paper discusses sixteen issues regarding MRSEVs and presents, in summary form, a proposed library of MRSEVs for 3-axis machining.

Funding for this work was provided to Catholic University by the National Institute of Standards and Technology under cooperative agreement Number 70NANB9H0923.

1 Introduction

1.1 MRSEVs

In machining a metal part, successive cutting operations are generally carried out, each of which removes material from the workpiece. A method of specifying what to remove in each operation is required. It is sometimes feasible to use features on the design of the part to determine what material to machine away in order to make the part. Very often, however, it will be desirable to machine the workpiece so that it has temporary features which do not appear on the design, or which do not correspond exactly to design features. To handle this situation, it is useful to have a method for describing the shape of material to be removed by machining operations which is independent of the design. Shapes of volumes to be removed are generated during process planning and used for NC-programming. The need for these material removal volume shapes was cited in a document, *NC Process Information Model* [Cain89], prepared for consideration in the IGES/PDES Organization (IPO) Manufacturing Committee. The term “Material Removal Shape Element Volumes” (MRSEVs) was defined in that document to refer to such shapes. That term, MRSEV, will be used in this paper.

Issues regarding MRSEVs are the focus of this paper. A library of proposed MRSEVs for 3-axis machining is presented in [Kramer92] and summarized at the end of this paper. A subset of the library has been implemented in an automatic NC-program generation system named OLPS [Kramer91a]. MRSEVs are used for passing shape information to OLPS in conjunction with process plans for machining.

1.2 Process Plans

Process planning may be defined as “a procedure for determining the operations or actions necessary to transform material from one state to another” [ANSI]. The core of a process plan is a set of procedures (steps), some or all of which must be carried out in order to achieve the objectives of the plan. The plan must also provide guidance on the alternatives for which steps must be carried out, and in what order. In simple plans this information may be implicit by the understanding that all steps will be carried out in the order listed.

A machining process plan is a plan for doing the machining required in a single fixturing of a part. It consists primarily of the specifications for a set of machining operations. The specification for a single cutting operation includes at least two types of data:

1. description of the operation - the type of operation, the specific type of tool to use, feed rate, spindle speed, horizontal stepover, vertical pass depth, and, possibly other information specific to the operation.
2. specification of the shape of the material to be removed - a pointer to a MRSEV.

1.3 Conceptual Model of Manufacturing

MRSEVs should be understood in the context of a particular conceptual view of manufacturing by machining. An activity and data flow model of the view taken here (excluding shop floor activities which would occur after the NC-program is written) is shown in Figure 1. The model is specific to cutting piece parts, in that it includes a workpiece, a fixture, and a setup. The model is not specific to any particular type of cutting, and could apply equally well to machining, turning, and most other types of cutting found in machine shops.

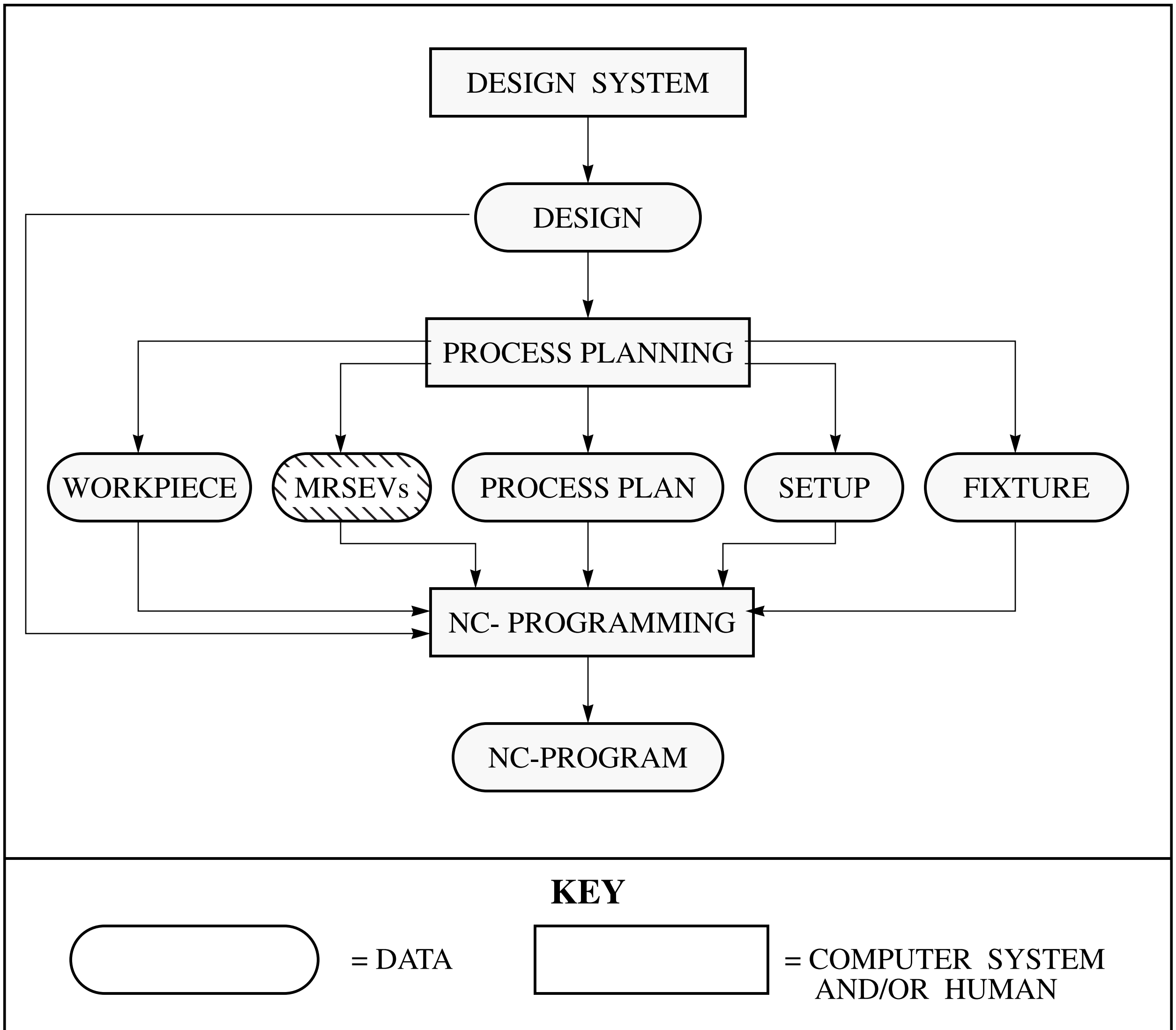

Figure 1. Manufacturing Model for Machining

In the model, a design system is used to produce a design. A design is taken to include specification of nominal shape as well as tolerances and surface finishes. The nominal shape may be expressed using a surfaced wire frame, a boundary representation, constructive solid geometry, form features, or even some other representation method. It may be useful to express the design in form features as the first step in process planning, regardless of how it comes out of the design system.

The process planning system takes in the design and puts out a machining process plan, a workpiece description, a set of MRSEVs, a setup description, and a fixture description. The setup description specifies where the workpiece, design, fixture and MRSEVs are with respect to a global coordinate system, which might be the coordinate system used to express the NC-program.

The NC-programming system takes in all of the output of process planning as well as the original design and produces an NC-program.

1.4 Related Work

In the document "NC Process Information Model" (NCPIM), the IPO Manufacturing Technology Committee created an information model for NC processes. The model was *"limited to the NC processes which are generated for removing material from mechanical piece parts"* [Cain89, p. 6].

The NCPIM defines the term Material Removal Shape Element Volume, as the *"Specified unit of volume of material to be removed in a material removal plan. It may include the removal of material from several regions of the part-in-process which are not adjoining."* [Cain89, p. 58].

The Computer Aided Manufacturing - International (CAM-I) organization has been studying many of the issues discussed here. Two of the most relevant CAM-I publications are [CAM-I85] and [CAM-I86]. Work on volume decomposition (for defining machinable volumes) has continued at CAM-I. The CAM-I literature, and several other researchers [Fridshal88] use the term "delta volume" to name the hunks of material being removed from a workpiece.

A method of defining "removing volumes" from a design for use in process planning was presented in [Li90]. That paper deals largely with feature recognition, a closely related area which we are not exploring here.

Additional related work is cited in discussion of specific issues.

2 MRSEV Technical Issues

Which issues are important depends on assumptions made about the situation. The division of responsibilities between process planning and NC programming is particularly important. It is assumed here that the process planner is aware of all relevant geometry in the machining environment, including the shapes of the final design, the workpiece at all stages of machining, fixtures, and tools. As compared with current industry practice for this division of responsibility, this paper puts relatively more responsibility with the process planner.

2.1 How Does a MRSEV Relate to a Machining Operation

Each metal cutting machining operation in a process plan will refer to a MRSEV. But how does the MRSEV relate to the operation? Should the MRSEV be exactly the volume cut away by the operation, should it be exactly the swept volume of the cutting tool, or what?

This issue has several sub-issues which are discussed in following sections. We propose that a MRSEV should relate to a machining operation as follows: the operation should accomplish a boolean subtraction (as normally defined - in chapter 9 of [Mortenson85], for example) of the MRSEV from the workpiece. Another way of stating that is: the volume described by a MRSEV should have no material in it when the machining operation is complete, and the operation should remove no material outside the volume.

Notice that we did not say that a MRSEV is the volume of material to be cut away by the operation. A MRSEV might not be totally within material. It may be argued that an additional constraint should be added requiring that the MRSEV be totally within material. This issue is discussed in section 2.6.

The actual swept volume of the cutting tool in carrying out a machining operation (after finishing its approach) may be exactly the same as the MRSEV, may be wholly contained in the MRSEV, or may intersect the MRSEV and extend outside it.

Although it is not required that the tool stay within the MRSEV during the operation, (only that it not cut any material outside of the MRSEV), if this requirement were added, MRSEVs could be used by the process planner to communicate information about where it is safe to run the tool, as well as information on volumes to be removed. Since all material in a MRSEV must be gone at the end of the operation, if the cutting part of the tool can be kept within the MRSEV plus an approach path known to be clear of the workpiece and fixture, it will be certain that the cutting part of the tool will not gouge the workpiece or collide with the workpiece or fixture. Note that an additional check for collisions of the shank of the tool, the tool holder, and the machine’s spindle with the workpiece and fixture would still be necessary to ensure safe machining.

2.2 Patterns, Groups and Replications

Should combinations of disjoint volumes be allowed as single MRSEVs? If so, what types of combinations are desirable? Should replications be allowed?

A pattern is a regular (by some measure) arrangement of copies of a basic shape. A group is a set of several shapes related by just declaring them to be related - a set of pockets of different shapes, for example. A replication is a reproduction of a previously defined shape in a different location and/or with a transformation applied to the shape. Mirroring is the most useful and common such transformation.

It is very desirable to provide for patterns, groups, and replications of MRSEVs.

By using a pattern of bolt holes as its MRSEV, for example, a single step in a process plan can give the instruction to drill them all. The most common patterns are circular and rectangular arrays. At least these patterns should be available for MRSEVs.

If three different pockets are all to be milled using the same tool, they can be made into a group, and a single process plan step can state that all three should be milled.

If a symmetric object is to be made (a drawer handle, for example), or if several identical objects are to be made from a single workpiece, replications are extremely useful.

The members of a pattern or group of MRSEVs may or may not be required not to intersect each other. There is no logical problem with allowing them to intersect, but keeping them disjoint will probably simplify machining them, and would forestall unsafe machining conditions, such as overlapping drill holes.

2.3 Relating MRSEVs to Workpiece Shape

In some schemes for describing the volumes to be considered in connection with machining operations, the volume is classified according to its relationship to the existing workpiece shape, as well as its own shape. Examples are provided by the systems described in [Anderson91], [Burd89], [Fridshal88], [Hummel91], and [Radack91]. Is this desirable?

The terms *slab* and *step* are typical names of volumes of this sort. In their simplest form, both *slabs* and *steps* are block-shaped solids. In their simplest applications, a *slab* is used to remove the entire top surface of a workpiece, as shown in Figure 2A, and a *step* is used to remove one side of the top of a workpiece, as shown in Figure 2B. This naming convention can be useful in process planning, since the name indicates what machining processes might be suitable (face milling for *slabs* and peripheral milling with an end-mill for *steps*, for example) but the use of *slab* and *step* to convey the relationship of the removed volume to the workpiece fails because the relationship cannot be precisely defined. For example, if an arm of the workpiece sticks up above the bottom of the *slab*, as shown in Figure 2C, is the *slab* no longer a *slab*? Certainly, if multiple nearby arms were crowded around the volume, as shown in Figure 2D, it would not be suitable to call the volume a *slab*, because the normal machining process used to make *slabs* (face milling) could not be employed since there would be no side access to the volume. When is a *slab* no longer a *slab*? It might be possible to develop an elaborate set of rules for deciding when a removal volume is a *slab*, but the gain does not seem worth the effort. Trying to define a *step* rigorously is an equally messy undertaking.

Because naming schemes of this sort fail in many instances, it does not seem practical to use them. Not using names to convey the relationship of removed volumes to workpiece shape makes the work of the process planner more difficult, but the difficulty shifts from trying to decide how to name a shape to trying to decide how to machine it, which is the key issue in the first place.

In the MRSEVs described in the remainder of this paper, no attempt is made to relate MRSEV shape to workpiece shape. This is intentional. Thus the volume that needs to be removed from a piece of stock to form what is commonly called an “exterior profile” is going to be a “pocket_with_islands”, and the common term “through hole” will not be found. These terms relate the removed volume to the shape of the workpiece. This does not mean, for example, that the distinction between a through hole and a blind hole is not important in machining, it only means that the distinction is not reflected in the description of the removed volume. Rather, it is intended that the distinction be embodied in the machining operation used to remove the volume.

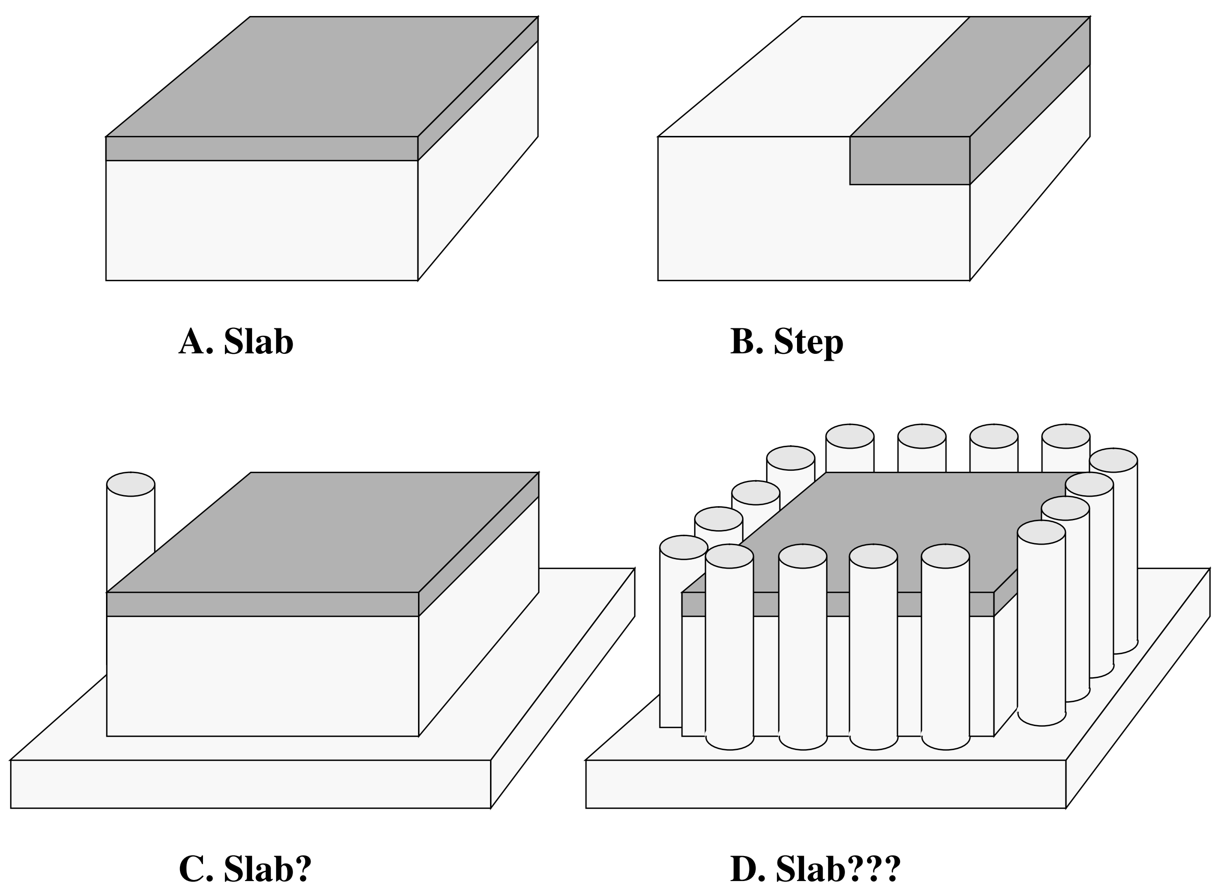

Figure 2. Slabs and Steps

2.4 Carrying Machining Information in MRSEVs

In some systems, machining information is carried in MRSEVs (called by other names). In the system described in [Graves88], the frame for a "hole" feature includes a slot named "manufacturing information" which contains machining parameters. Another example is provided in [Hummel91], which includes features named "groove-requiring-special-tooling" and "hole-with-large-length-to-diameter-ratio". Is this desirable?

Usually, embodying machining information in a MRSEV is done by attaching the name of an operation to the name of the MRSEV type. For example, the volume to be removed by a drilling operation might be called a "drill hole", whereas the same volume might be called a "bore hole" if it is used with a boring operation. Putting machining information into MRSEVs this way can be useful if the MRSEVs are created before process planning is done, as in design-by-feature systems. As discussed earlier, however, this approach is often not adequate to support manufacturing. Because we are assuming here that the process planner is the one who defines the MRSEVs, there is no advantage to be gained by attaching machining information to them, so we are not doing it. Rather, we assume that libraries of machining operations and MRSEVs will be matched appropriately but defined independently. An example of a library of machining operations, the one used in OLPS, is given in [Kramer91b]. That library is built on top of a general-purpose process planning language named ALPS [Ray91].

In OLPS, a "hole" MRSEVs might be used with five different operations: twist-drill, center-drill, counterbore, rough-mill, and finish-mill.

2.5 Fixed Library vs Define as Needed

Should the MRSEVs be from a fixed library of canonical types or should they be defined as needed?

A library of canonical types may be composed of parametrically defined features, where the parameters represent relatively high-level information about shape, such as a single radius for all corners of a pocket. The advantage of a fixed library is that it is feasible to write computer-executable algorithms which will automatically generate tool paths for cutting out the volumes in a fixed library. The algorithms are parametric and use the feature data fairly directly. Another advantage is that parametric features are relatively easy to define. The disadvantage of the fixed library is that there is inevitably a shape to be cut which cannot be described in terms of anything in the library, so that human intervention will be required to generate a cutter path.

If shapes are to be defined as needed, the most straightforward method that will permit almost any shape to be expressed is to use a boundary representation (B-rep). Automatic generation of tool paths directly from a B-rep is much more difficult, because the connection between B-rep data and cutter path generation is not direct. Defining B-rep solids is also more difficult and generally requires a solid modeler.

A compromise position is to have a core library of parametrically defined features and allow the user to add to the library as needed. If additions to the library are restricted to shapes which can be defined with a formal description language in terms of the core library, it may be feasible to handle the resulting shapes automatically. That still leaves the problem of shapes which are not expressible using the expanded library. This problem might be handled by allowing MRSEVs which cannot be expressed using the library to be expressed in B-rep form as a collection of faces. Today's automatic tool path generation systems cannot deal with shapes cobbled from an arbitrary collection of B-Rep faces, and systems available in the next few years are not likely to, either. Thus, tool path generation systems which allow such shapes will not be fully automatic for the foreseeable future. Rather, such systems will need to be able to call for manual intervention when a MRSEV expressed in B-rep form is encountered.

In any event, it seems desirable to start with a fixed library.

2.6 Is Air Allowed in a MRSEV?

Consider a machining operation that references a MRSEV. In all cases no material should be left inside the MRSEV when the operation is finished. However, there are two choices about what the MRSEV encloses before the operation starts:

1. The MRSEV may enclose both air and material.
2. The MRSEV must enclose only material.

Which of these is best?

As discussed below, it seems desirable to allow air in MRSEVs. A simple part which illustrates this issue is shown in Figure 3.

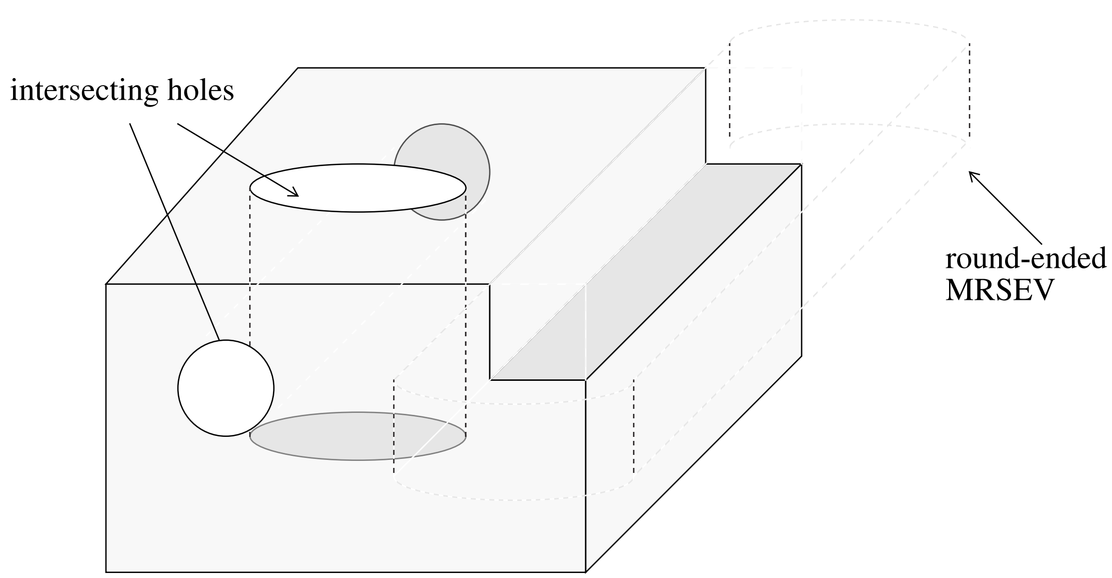

Figure 3. Simple Part with MRSEVs

2.6.1 Air Allowed

If air is allowed in MRSEVs, it is feasible to handle a much wider range of parts with a given library. This is because MRSEVs which intersect and a MRSEV which extends outside the workpiece may produce complex geometry. If a MRSEV can include air, there is no requirement that the complex geometry be representable using the library. If the MRSEV cannot include air, the complex geometry must be represented. In Figure 3, for example, suppose the hole in the side of the block has been made before the hole in the top, and the hole on the top intersects the side hole. If air is allowed in the MRSEV, the hole in the top can be represented as a cylindrical volume, and the hole in the side ignored. If no air is allowed, the MRSEV must be the difference of two cylinders.

If air is allowed in MRSEVs, an unsophisticated tool path generation system can do an adequate job by producing a tool path for each MRSEV which sweeps out the entire volume, including any air, while a sophisticated system can run a geometric modeling process as it generates tool paths and cut a minimum of air by analyzing the geometry of the situation.

Consider, for example, the step on the upper right side of the block. The step may be produced by sweeping an end mill through the round-ended volume shown with a dotted outline. If that volume sticks out of the block farther than necessary, an intelligent system could take a shorter cut than a dumb one.

As mentioned earlier, allowing air in the MRSEV gives the process planner a means of telling the NC programmer where it is safe to run the tool when it is outside the workpiece. The step in Figure 3 made by the round-ended MRSEV is an example of this.

Of course a smart system would be able to determine where the swept volume of the tool could pass outside the MRSEV, even if the MRSEV is allowed to contain air.

If air is allowed in MRSEVs, in order to find the actual shape of the workpiece when a MRSEV is removed, a solid modeling system will be required to do a boolean subtraction of each MRSEV from the workpiece.

2.6.2 Air not Allowed

The system described in [Fridshal88] is an example of a system in which the material removal volumes, called “delta volumes”, may not contain air.

If MRSEVs must be all material, the cutting tool must sweep the entire volume of the MRSEV. When cutting an airless MRSEV, it will often not be possible to keep the tool inside the volume while cutting it (as in cutting the step shown in Figure 3, where the exact volume to be removed has sharp corners into which no cutting tool will fit). Thus it will be necessary to have a system which is intelligent enough to determine where it is safe to pass outside the volume being removed.

If MRSEVs must be all material, it must be possible to express complex geometry, the intersection of the two cylinders in Figure 3, for example. The only way a fixed library of MRSEVs is likely to be usable for describing geometries of intersecting MRSEVs is if boolean subtraction may be used with the library. A solid modeling system would be required to do this subtraction.

Another drawback of not allowing air in MRSEVs is that simply defining the shape of the MRSEV is likely to determine the order in which machining operations may be done. In the example of the two intersecting holes, the first MRSEV to be cut will be a solid cylinder, while the second will be a cylinder with a hole through it. In order to avoid cutting air, the first one will have to be cut first. If air were allowed, deciding which to cut first could be deferred until later.

2.7 Linking MRSEVs to Design Data

How can MRSEVs be linked to design data?

In order to have a manufacturing process that can learn from past mistakes, it will be necessary to have feedback to manufacturing from inspection and testing (not shown on Figure 1). One critical piece of information for machined parts will be knowing what machining operation or operations formed each feature of the finished part. Without this information, if some feature is out of spec, it will not be possible to determine why and correct the problem. Thus, it is necessary to link process plan steps or MRSEVs to design data. This issue has not been addressed in the library given here or in OLPS. Future development of MRSEVs will need to deal with this issue.

It is anticipated that linking MRSEVs to design data will not require any new shapes to be added to the library. Rather, each process plan will include a pointer to a design and an attribute will be added to each metal-cutting step of the plan (or to each MRSEV) indicating what portion of the design is produced by carrying out the step (or by machining the MRSEV).

2.8 Closed vs. Open MRSEVs

Should MRSEVs be defined as closed volumes or as open ones, or should both alternatives be allowed?

The idea of a closed MRSEV is elementary. Any solid physical object occupies a closed volume.

A MRSEV can be left open in several ways and still express clearly what volume is to be removed. If a cylinder of material whose axis is parallel to the z-axis of a machining center is to be removed from a part, for example, no top is required for the cylinder in the plus-z direction. It may simply be taken as a convention that cutting will start at the top of the workpiece, wherever that may happen to be. Similarly, if a through hole is made, the cylinder might be open on the bottom. A third type of open volume is open on the sides; the idea being to remove all material from a workpiece to a given depth which is outside of a given profile.

It is useful for a MRSEV to be closed in order to inform the NC programmer where it is safe to run a tool. If open MRSEVs are used, the NC programmer has more work to do to determine where it is safe to run the tool.

If MRSEVs are to be used by solid modelers, it will be easier if they are closed. Some modelers will not handle open volumes.

If MRSEVs are to be drawn, it will be easier if they are closed.

In view of the foregoing, it seems slightly preferable to use closed MRSEVs only.

2.9 Complete Definition vs. Ambiguity

Should MRSEVs be defined entirely unambiguously, or is some ambiguity desirable?

The initial inclination of most people is to say: Let's stay out of trouble - no ambiguity allowed. This inclination usually disappears as soon as threads are considered. Almost no one wants to specify thread timing. Normally, it is of no functional relevance in a design, and it is a nuisance to accomplish on a machining center, so it is ambiguous in almost all specifications.

Some other types of ambiguity may be desirable, as well. A prime candidate is the location of the top of a closed MRSEV. In most cases, there will be no material above the top of a MRSEV, so the location of the top of the MRSEV is not critical (assuming air is allowed inside MRSEVs). It may simplify the job of the NC programmer if the top of the MRSEV is left undefined. Similarly if a MRSEV is known to extend outside the workpiece on all sides, the sides of the MRSEV could be left undefined. The ambiguity could be implemented in the case of the top of the MRSEV, by making the depth of the MRSEV an optional attribute, whose omission in any instance would mean "the top of this MRSEV is anywhere above the workpiece". Leaving the sides of a MRSEV ambiguous could be implemented by making the data which tells where the sides are optional. Again, if this data were omitted, it would mean "the sides of this MRSEV extend beyond the workpiece". No ambiguity of this sort is provided in the library described in section 3, but it might be a desirable addition, and eliminate the need for what is nearly meaningless data.

One other type of ambiguity appears to be desirable and is proposed in the library. That concerns the shape of the end of an open edge_cut MRSEV.

2.10 Applications Supported

The point of having MRSEVs is to provide a method of defining volumes that can be removed by machining operations. The main question here is, what machining operations should the MRSEVs support? A secondary question is: Once we have defined MRSEVs, are there other uses to be made of them, beyond what they were defined for?

2.10.1 Machining Operations Supported

However a given library of MRSEVs is defined, the library should provide sufficient shapes to support some basic collection of operations on some class or classes of NC-machines.

Within NC machining there is a lot of choice. Turning and milling for example, involve quite different shapes. Within milling, the distinction between 2.5-axis, 3-axis, 4-axis, and 5-axis is important, with an increasingly complex set of MRSEVs required. For 5-axis milling, for example, it might be desirable to include a pattern in the library for making a ring of holes around a cone, with each hole perpendicular to the surface of the cone. A 3-axis machine cannot drill such a pattern in a single fixturing, so it would not be desirable to put that pattern into a library of shapes for 3-axis milling.

The library described in section 3 is intended to support 3-axis milling. This library should be useful as the core of a library for 4-axis or 5-axis milling, since any shape producible on a 3-axis machine can be produced on one with 4 or 5 axes.

This paper does not address the issue of MRSEV libraries for other types of NC machines.

2.10.2 Other Uses

MRSEVs might be used to describe the shape of the entire machining environment, so that MRSEVs could be used exclusively as input to a geometric modeler. This means using MRSEVs to represent positive volumes, and not just voids. Each MRSEV is defined as a closed volume. We have been thinking of MRSEVs as volumes which are voids, but we could change our thinking to allow a MRSEV to be either a void or a hunk of material. The change in thinking is easy to implement. One method is to simply add a flag to each MRSEV indicating whether it is material or void. Keeping lists of material MRSEVs separate from lists of void MRSEVs would be another method.

However, if MRSEVs are to be used to describe the entire environment, they must represent workpieces, fixtures, and tools as well as volumes to be removed. It is unlikely that the shapes of these things would correspond well to the shapes of removable volumes. Thus the representation of workpieces, fixtures, and tools would probably have to be approximate, if given using MRSEVs.

It will probably be preferable to use B-Rep for workpieces, fixtures, and tools, and convert MRSEVs to B-Reps before handing them to a solid modeler. In this way the modeler could deal entirely with B-Reps (which many modelers are designed for).

2.11 Tolerance Information

The definition of MRSEVs must be compatible with an approach to machining that insures that parts will be made within the tolerances specified in the design. How can this be done?

2.11.1 Tolerance Information in Process Plans

Tolerance information might be embodied in a machining process plan without tolerancing MRSEVs as follows.

First, we arrange machine tools in classes and the require that each machining process plan specify the class to be used.

Second, we require that machining operations included in a process plan be drawn from a predefined library.

Then we may provide for tolerances in a process plan step (without putting tolerance information into MRSEVs) either

1. By putting a tolerance attribute in the step, or
2. By specifying that a known level of tolerance will be held when carrying out a named operation on a machine in a specified class, or
3. By including in the step a pointer back to the design features made by the step.

Hybrids of these methods (1&2, 2&3, or 1&2&3) might be desirable.

If method 2 is used, the NC-programmer must be aware of the level of tolerance expected from each machining operation in the library but does not need to be aware of tolerance requirements in the design.

Method 3 is not sufficient by itself since some machining operations will make temporary features that do not appear in the design. Some level of tolerance will be required of these temporary features. The tolerance zone of the wall of a rough-cut pocket must not be allowed to extend beyond the tolerance zone of the finish-cut wall, for example.

2.11.2 Tolerance Information in MRSEVs

As an alternative or adjunct to putting tolerance information in process plan steps, tolerance information could be put into MRSEVs themselves by attaching tolerances to some of the attributes that define each MRSEV.

Not all tolerance requirements that might be present in a design can be represented by tolerancing MRSEV attributes, however, so putting tolerances only on MRSEV attributes will not be sufficient in some (possibly most) cases.

More elaborate schemes for tolerancing MRSEVs than tolerancing attributes might be devised, but it seems unlikely that this would be preferable to simply pointing back to the design.

2.11.3 Nominally Exact vs. Nominally Inexact Machining

Some shapes can be cut nominally exactly (i.e., if the machine and cutter were perfect, the exact shape would be cut). For example, a cylindrical hole can be made nominally exactly by drilling with a drill whose diameter is the same as that of the hole. Other shapes, for example a depression shaped like the bottom half of a football lying on its side, can nominally be cut only approximately. The normal method of cutting the football-shaped depression is to rough cut the depression and then finish cut it with a ball-nosed end mill whose radius is smaller than the minimum radius of curvature of the depression, making passes with the end-mill which are close enough together so that all points on the nominal machined surface are within some designated tolerance of the intended surface. The actual machined surface is not as close to the intended surface as the nominal machined surface because of normal machining factors.

If process plan steps carry tolerance information by any of the three methods outlined above, no tolerance specification is required in a MRSEV for shapes which can be nominally exactly machined. However, if only method 2 were used, in the case of nominally inexact machining (such as the football), it would be desirable to put tolerance information into the MRSEV which specified how far from exact the nominal machined shape could be.

2.11.4 Recommendation

In view of the foregoing discussion, it seems desirable to let the process plan carry most of the burden of specifying tolerance information and to put little into MRSEVs. The library presented in section 3 provides for tolerance information only in the case of MRSEVs which cannot generally be made nominally exactly.

2.12 Using MRSEVs for Operations on Surfaces

Can MRSEVs be defined to be usable with operations that act on surfaces?

2.12.1 Thin Layer Removal

MRSEVs can be used with machining operations which remove only a small layer of material from a workpiece. In OLPS, “finish-mill” is such an operation. A thin layer is not explicitly represented for this purpose. Rather, a MRSEV is defined which represents the volume to be void when the finish-mill is completed. It is assumed that most of that volume will already be void before the operation starts, so that only a thin layer needs to be milled away. If the shape being made is a pocket, for example, practice in OLPS has been to define two pocket MRSEVs, one for rough cutting and a slightly larger one of almost the same shape for finish cutting.

This procedure works in OLPS, but the finish milling is inefficient in the case of pockets whose sides or bottom are in air; a lot of useless cutting through air is programmed by OLPS in such cases.

The inefficiency could be eliminated if a solid modeler were used by the tool path generator to determine where material is and only cut in those places. This is a difficult task.

The inefficiency could be lessened by splitting the finishing operation into two parts, the first to finish the sides of the pocket, and the second to finish the bottom. One of the two operations has to take out the material where the sides meet the bottom. It seems practical to have this done when the sides are finished. If this were done, it would seem preferable to have separate MRSEVs for the two operations, since the tool should avoid the sides while finishing the bottom. The MRSEV for the bottom could be a very shallow pocket, whose depth is equal to the thickness of material to be removed. The

MRSEV for the sides could be the entire pocket to be finished. The MRSEV for the sides could also be a groove equal in width to the diameter of the tool, which passed around the sides of the pocket.

All of these methods for finishing a pocket using volumetric MRSEVs seem either inefficient, difficult, or awkward. The problem might be alleviated by introducing a new class of MRSEVs which are descriptions of thin sheets, rather than volumetric solids. A sheet MRSEV would be described by a description of a surface patch and a thickness. Sheet MRSEVs would be used for conveying the shape information required by finishing or other surface operations.

This idea has not been pursued any further than given here. It deserves further study.

2.12.2 Other Surface Features

There is a large class of surface features which can be produced on a machining center. The only one of these which is covered by the MRSEV library is internal threads. A great deal can be done with text on a machining center which cannot be effectively represented using boundary representations (machinable characters, fonts, spacing, placement, and text transformations). Variable depth line drawings can also be cut on a machining center but not easily represented using boundary representations. Neither text nor variable depth line drawings are covered by this MRSEV library. Groove MRSEVs, however, can be used reasonably efficiently to cut constant depth line drawings. Discussions of feature representations for text, constant depth line drawings, and variable depth line drawings may be found in [Kramer88], [Kramer89a], and [Kramer89b].

2.13 Range of Parts

What range of parts should be producible using a MRSEV library?

The MRSEV library described here is intended to be adequate to make parts which can be described in a boundary representation in which the surfaces are elementary surfaces (plane, sphere, cone, cylinder, torus) and which can be machined on a 3-axis machining center using cutting tools whose spun volume (the swept volume of the tool when it is spinning but not feeding) has elementary surfaces where it contacts the workpiece.

It is expected that this library should be adequate to deal with a large fraction of the piece-part work found in most machine shops which are machinable on a 3-axis machining center. The library is not usable for dealing with sculptured surfaces. For the aerospace industry, which deals with many parts that have to fit into the generally non-rectilinear shape of an airframe, a smaller fraction of parts can be expected to be covered than for an industry such as construction machines.

2.14 MRSEVs and NCRSEVs

How do MRSEVs relate to NC Removal Shape Element Volumes (NCRSEVs)?

The NC Process Information Model defines NC Removal Shape Element Volumes as well as MRSEVs. Each NCRSEV is *a decomposition of a material removal shape element volume into NC removable shape element volumes. Each volume is entirely contained within its associated material removal shape element volume* [Cain89 - page 59].

The NCPIM does not go into further detail about the two types of volumes. One example of the volumes is given: a Material Removal Shape Element Volume could be a series of holes to be made by a single drill [Cain89, p. 58]. It seems intended that an NC Removal Shape Element Volume must be all one piece, and not broken up into parts.

The MRSEVs library defined in section 3 provides for the relationship between MRSEVs and NCRSEVs suggested in the NC Process Information Model.

If a MRSEV includes only one elementary_volume and has a copy_maker, then each copy of the elementary_volume would normally be expected to be an NCRSEV. A prototypical example is that if a circular pattern of bolt holes is defined as a single MRSEV by defining one hole and a circular array, then each of the implicitly defined holes in the MRSEV would be an NCRSEV.

If a MRSEV includes more than one elementary_volume, and the volumes intersect or have adjoining faces, then it might be useful to make the union of the elementary volumes be an NCRSEV. For example, if a MRSEV consists of a nest of three pockets, and a step from a process plan says to rough-mill the MRSEV, then it might be a good machining tactic to combine the three pockets into a single NCRSEV before generating a tool path to cut them.

It is not clear that a standard for NCRSEVs will be required, because it seems likely that NCRSEVs will be used entirely within NC-programming, not shared with other modules.

2.15 Coverage of Common Variants

To what extent should specialized common shapes be included in a MRSEV library?

Certain two-dimensional forms appear commonly on machined parts as either profiles of pockets or cross sections of grooves. L-shapes, and T-shapes are two of these. It would be feasible to define MRSEVs which embody these shapes - an L_pocket, for example.

The library defined here provides relatively few MRSEVs of these specialized types. Most of the shapes provided by these specialized types can be created with the library defined here by using a more generalized MRSEV type. An L_pocket, for example, can be formed as an other_pocket MRSEV in the library.

The specialized shapes could be defined parametrically, thus using much less data than required for the more generalized types. An L_pocket, for example, would require only four attributes (base_length, base_width, ascender_length, and ascender_width comprise a workable set) to define its outline, excluding any corner rounding.

It may be desirable to add these specialized types to the library defined in section 3.

2.16 Accessibility for Machining

To what extent should MRSEVs carry information about their accessibility for machining?

In the manufacturing model of Figure 1, it is not necessary for MRSEVs to carry information about their accessibility for machining. It is expected that the process planner will have determined accessibility and specified the order of operations in the process plan so that each MRSEV will be accessible when the operation to make it is performed. Thus, the MRSEV library described here has no place to put information about accessibility.

During process planning, however, it is desirable that alternatives be considered for how to machine the part, in order to build a reasonably efficient plan. At that stage, several different sets of MRSEVs might be generated and alternative plans made for machining them and then compared. It is very useful for access information to be attached to material removal volumes in order to build the alternatives. Some systems, [Karinthi92], for example, provide accessibility information for this purpose.

If MRSEVs are used for optimizing process plans as just described (which seems like a good idea, although it has not been attempted by the author), then it seems desirable to make a place for accessibility information in the definition of the MRSEV library.

3 A Library of MRSEVs for 3-Axis Machining

This section defines, in summary form, a library of MRSEVs for use with process plans for a 3-axis machining center. This could be either a vertical-spindle machine or a horizontal-spindle machine. Both types typically perform the same operations. A much more detailed description of the library is given in [Kramer92]. The library takes into account the resolution of most of the issues discussed in the previous section.

3.1 Nature of MRSEVs in Library

At the elemental move level, the more capable 3-axis machining centers can drive the tip of a cutting tool in only three distinct trajectories: (i) a straight line between two points in 3-dimensions, (ii) an arc of a circle (or a complete circle) whose axis is parallel to one of the three principal axes of the machine, or (iii) an arc of a helix whose axis is parallel to one of the three principal axes of the machine. The library includes MRSEV types (holes, ramps, and grooves) for representing the swept volumes created when a cutter whose spinning shape is a cylinder with either a flat end or a rounded end is driven along any of those three trajectories - with the top of the swept volume cut off flat, and with the limitation that the axis of a circular or helical arc must be parallel to the axis of the cutter.

At a higher level, certain more complex shapes are commonly found on machined parts. These include complex grooves, pockets, steps, and exterior profiles. The library provides shapes for all of these. As stated earlier, however, it is intended that the “pocket” MRSEVs defined here be used for producing not only what is usually called a pocket, but also for producing what are usually called steps and exterior profiles and for removing what are usually called slabs.

All the MRSEVs are to be construed as closed volumes. The tops of all MRSEVs are to be construed as being planar.

Each type of MRSEV has a native coordinate system attached to it. When an instance of a MRSEV is built, the native coordinate system is located with respect to some parent coordinate system by a locator.

Most of the shapes in this library are similar to shapes provided in the STEP Form Features Information Model [Dunn92]. The relationship between that model and this library is explored in detail in [Kramer92].

3.2 Description Methods

The library of MRSEVs is described here as a hierarchy of entity types. Each entity type has a number of attributes. The attributes have values which must be of some specific data type (single instances or sets of: a real number, a string, a choice of one of a set of tokens, or an entity type). Some attributes are optional. An entity may have subtypes. The subtypes inherit all the attributes of the parent types and may have additional attributes of their own. Only entity types which do not have subtypes can be used as

MRSEVs referenced by process plans. This description method is conceptually identical to a subset of EXPRESS [Spiby91], but we are not using EXPRESS syntax in this section, and any language that includes the elements just described could be used.

Many of the shapes in the library are defined with the aid of a "profile". A profile is an ordered collection of arcs of circles and straight line segments lying in a plane. The elements are joined end-to-end to form a continuous curve. Profiles are used in the library both as two dimensional shapes which may be swept or rotated through space to make a solid, and as paths through space along which other profiles or cross sections can be swept. Subtypes of profile are defined in [Kramer92].

3.3 Hierarchy Figures

Figure 4 shows the main hierarchy of MRSEVs in the library. The following conventions are used:

1. Each entity type is defined in a white rectangular box.
2. The names of entity types are given in **boldface**.
3. The names of attributes of entities types are given in *italic*.
4. Select data types are given in helvetica.
5. The data type of each attribute is given after the attribute name. If the data type is not real, string, or select, it must refer to the name of an entity type defined elsewhere. Only one entity type referenced in this way is defined on Figure 4; that is mrsev_volume, which is the type of one of the elementary_volumes of a mrsev. That connection is shown near the top of the figure by a heavy broken line.
6. Parent entity types are located above their subtypes and are connected to their subtypes by solid lines, which may branch horizontally. A complete set of all attributes for each instantiable (leaf node) entity type in the library can be extracted from Figure 4 by tracing these lines upwards through all the higher ancestors of the entity type.

Figures 5 and 6 use the same graphic conventions. Figure 5 shows the hierarchy of islands which may occur in pockets. Figure 6 shows the hierarchy of copy methods which may be used with MRSEVs.

3.4 The Main Hierarchy

As shown in Figure 4, the topmost entity of the main hierarchy is mrsev. A mrsev has three attributes: (i) *elementary_volumes*, the value of which is a set of one or more mrsev_volume, (ii) *copy_maker*, the value of which is an optional mrsev_copy_method, and (iii) *mrsev_name*, the value of which is a string. Within a physical file defining a set of mrsev's, no two mrsev's should have the same *mrsev_name*. In the remainder of this paper, this specific entity (mrsev) will be spelled using lower case letters, while the generic MRSEV will continue to be spelled in capitals.

Mrsev_copy_methods are discussed in subsection 3.6

The rest of Figure 4 shows the subtypes of mrsev_volume. The two main subtypes are primary_mrsev_volume (which has a location of its own) and secondary_mrsev_volume (which has location only with respect to a primary_mrsev_volume). The only subtype of secondary_mrsev_volume in the library is thread, which is a right-handed thread on the interior of a hole (for holding a bolt).

Every primary_mrsev_volume is located in the overall coordinate system for MRSEVs (mentioned earlier) by the *location* attribute, the value of which must be an “axis_placement”. This indicates where the native coordinate system of the MRSEV is with respect to a global coordinate system.

The subtypes of primary_mrsev_volume and their approximate shapes are as follows. In most cases there are further restrictions on the shapes so that the shapes can be produced by machining. Each shape is bounded on one side by a plane which forms the upper surface of the mrsev_volume and lies perpendicular to the axis of the cutting tool when the shape is in position to be machined. Details of the location of the upper bounding plane are not given in the descriptions which follow.

1. groove - the shape resulting from sweeping a nice_closed_profile along a general_profile path while keeping the nice_closed_profile perpendicular to the path. Typically, the nice_closed_profile will be the silhouette of a cutter and the path will lie on a flat surface of a workpiece, so that removing this shape from a workpiece results in what is commonly called a groove.

2. ramp - the shape resulting from sweeping a cutting tool either (i) in a straight line which does not lie in the plane perpendicular to the axis of the tool or (ii) in an arc of a helix whose axis is parallel to the axis of the tool.

3. linear_sweep - the shape resulting from sweeping a nice_closed_profile along a straight line perpendicular to the plane of the profile. Typically, the line will be perpendicular to a flat surface of a workpiece. The two subtypes of linear_sweep are holes and pockets. Islands of material may be left in some types of pockets. Islands are discussed in section 3.5.

4. rotation_pocket - the shape resulting from swinging a nice_open_profile all the way around an axis that passes through the ends of the profile. For a horizontal_rotation_pocket, part of this shape is cut off with a plane that is parallel to the axis.

5. edge_cut - the shape resulting from sweeping a right triangle along a general_profile path while keeping the triangle perpendicular to the path. The hypotenuse of the triangle may be replaced by an arc of a circle. The use of a MRSEV of this type is in flattening or rounding an edge of a workpiece.

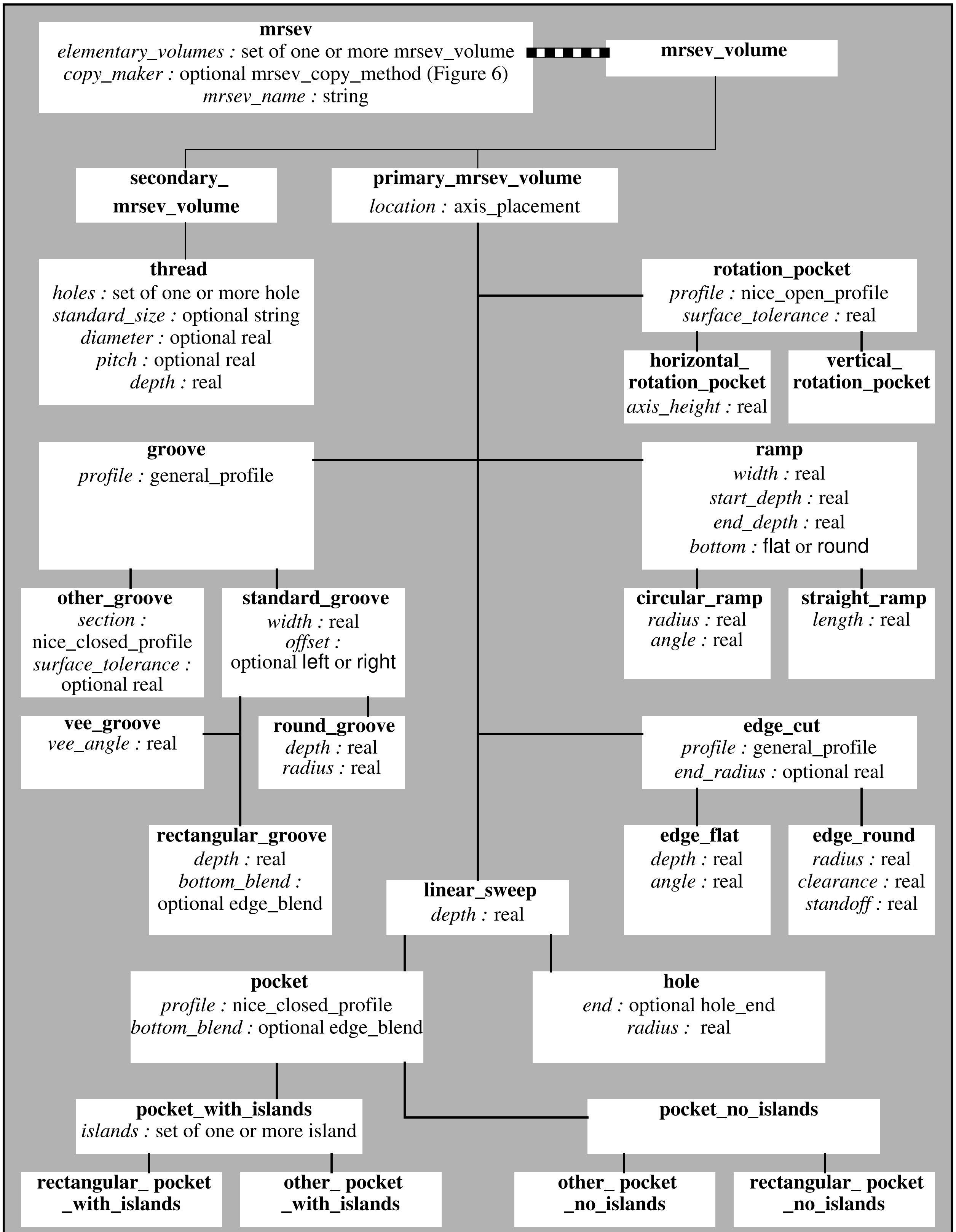

Figure 4. Main Hierarchy of MRSEVs in Library

3.5 Islands

An island is a closed volume which is to be boolean subtracted from a pocket MRSEV. Only the (flat-bottomed) pocket_with_islands subtype of pocket may contain an island. Because an island is boolean subtracted from its parent pocket, if an island extends through the top surface of the pocket, that portion of it which is outside the pocket has no effect; a tall island does not add material to a workpiece.

As shown in Figure 5, three subtypes of island are defined. Each may best be thought of as the opposite of one of the MRSEV types. The island is what you would get if you filled its opposite with material (ignoring edge conditions), as if casting in a mold, and then inverted the mold, extracting the cast shape.

1. mesa_island - the opposite of a pocket without islands. A mesa_island looks like a mesa.
2. anti_groove_island - the opposite of a groove.
3. rotation_island - the opposite of a rotation_pocket.

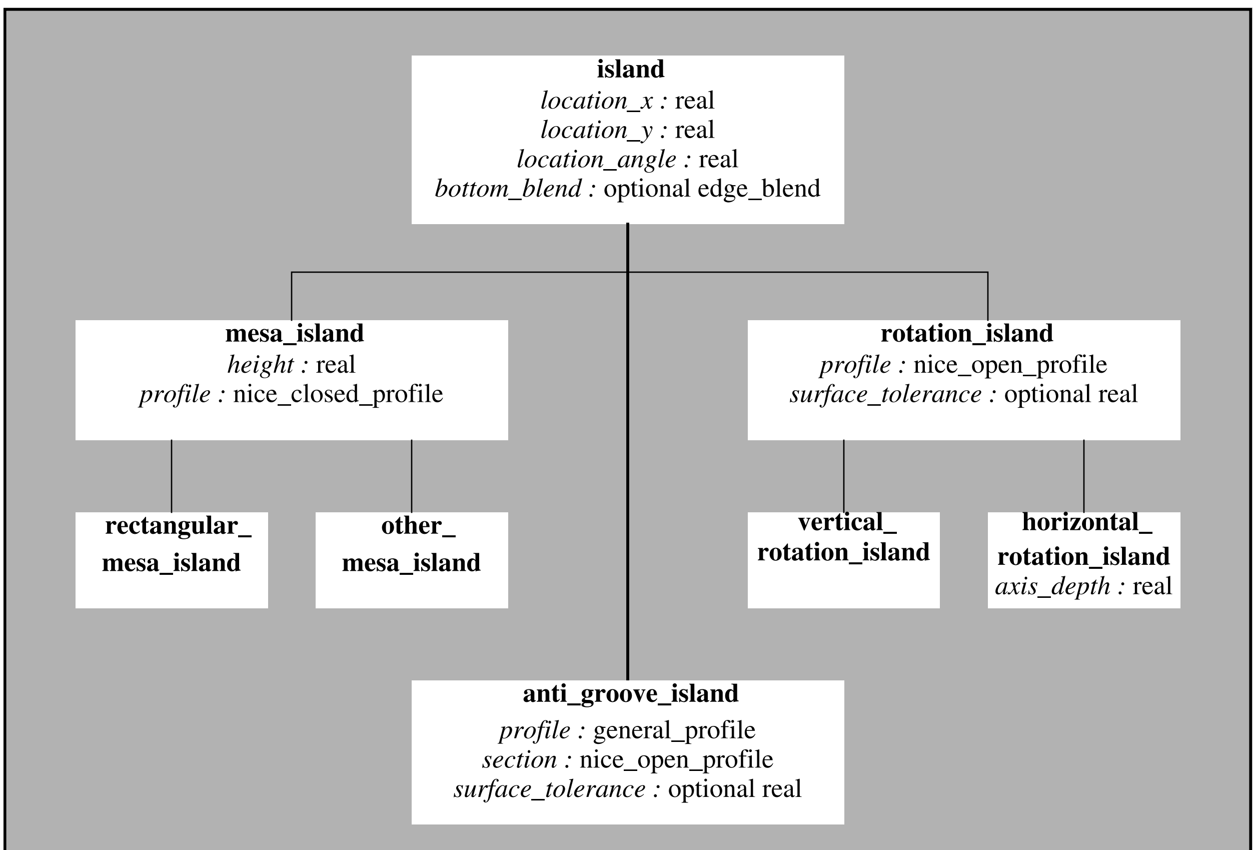

Figure 5. Island Hierarchy

3.6 Groups and Copy Methods

Groups are provided in the MRSEV library by having the value of the *elementary_ volumes* attribute of a mrsev be a set of mrsev_volume's rather than a single mrsev_volume. This is shown in the top block of Figure 4. Of course the set may (and typically will) have only one element. The mrsev_volume's in a set do not need to be of the same type. For example, pockets and grooves might both be included.

Copy methods include patterns and replications. Figure 6 shows the hierarchy of copy methods.

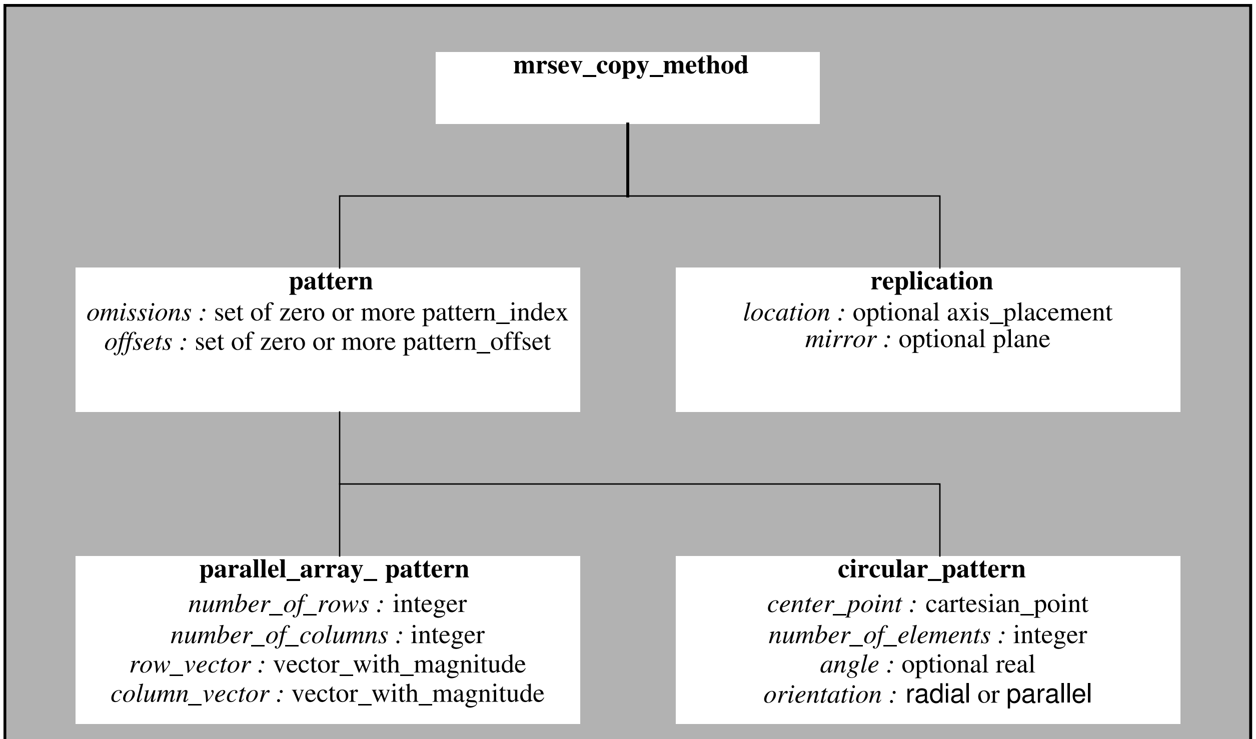

Figure 6. MRSEV Copy Methods

A pattern is a rule for making additional copies of some base shape or shapes. All patterns in this library are two-dimensional and are to be interpreted as applying in the xy-plane of the MRSEV file. The pattern does not include any base shape. The native coordinate system of the pattern is the coordinate system of the MRSEV file.

4 Conclusion

This paper has discussed sixteen issues regarding MRSEVs and proposed resolutions for most of them. It is interesting how rich the problem of deciding how to machine a part becomes as one gets closer to it. The paper has also presented a library of MRSEVs for 3-axis machining, a significant portion of which has been implemented in an automatic NC-program generation system, OLPS.

The paper has not discussed methods of generating MRSEVs suitable for use in a process plan for machining a particular design. That is an extremely important, but separate issue, which deserves attention by scholars and systems developers in the field.

The paper has also not discussed libraries of MRSEVs for other than 3-axis machining. While the issues covered in this paper apply to MRSEVs of any sort, there may well be additional issues that would emerge from attempts to build other libraries.

References

[Anderson91] Anderson, D., A Quick Turnaround Cell, proceedings of a *Workshop on Feature-Based Design and Manufacturing*, sponsored by Martin Marietta Energy Systems, held in Gatlinburg, Tennessee, June 1991

[ANSI] American National Standards Institute, Industrial Engineering Terminology, Production Planning and Control, ANSI, 1973, p. 16

[Brooks87] Brooks, S. L., Hummel, K. E., and Wolf, M. L., XCUT: A Rule-Based Expert System for the Automated Process Planning of Machined Parts, proceedings of ASME Annual Meeting, ASME, 1987, PED-Vol 25, pp. 181-194

[Burd89] Burd, W. C., The Milling Assistant, Sandia Report SAND89-1519, Sandia National Laboratories, Albuquerque, New Mexico, September 1989

[Cain89] Cain, W. D., NC Process Information Model Version 1.0, unpublished, February 1989

[CAM-I85] CAM-I, Volume Decomposition Algorithm, Document Number R-85-ANC-01, CAM-I, Arlington, Texas, 1985

[CAM-I86] CAM-I, Part Features For Process Planning, Document Number R-86-PPP-01, CAM-I, Arlington, Texas, 1985

[Dunn92] Dunn, M., Industrial Automation Systems - Product Data Representation and Exchange - Part 48: Integrated Generic Resources: Form Features, ISO TC184/SC4 Document N102, January 1992

[Fridshal88] Fridshal, R., Features at General Dynamics, proceedings of a *Workshop on Features in Design and Manufacturing* held at UCLA, February 1988

[Graves88] Graves, G. R., et al, An Interface Architecture for CAD/CAPP Integration Using Knowledge-Based Systems and Feature Recognition, *International Journal of Computer Integrated Manufacturing*, Vol. 1, No. 2, pp. 89-100

[Hummel91] Hummel, K. E., Some Important Issues in the Representation and Recognition of Manufacturing Features, proceedings of a *Workshop on Feature-Based Design and Manufacturing*, sponsored by Martin Marietta Energy Systems, held in Gatlinburg, Tennessee, June 1991

[Karinthi92] Karinthi, R. R., and Nau, D., An Algebraic Approach to Feature Interactions, accepted for publication in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992

[Kramer88] Kramer, T.R. and Jun, J., The Design Protocol, Part Design Editor, and Geometry Library of the Vertical Workstation, NBSIR 88-3717, January 1988.

[Kramer89a] Kramer, T.R., Contour Outlines, NISTIR 4586, National Institute of Standards and Technology, October 1989

[Kramer89b] Kramer, T.R., Enhancements to the VWS2 Data Preparation Software, NISTIR 89-4201, National Institute of Standards and Technology, November 1989

[Kramer91a] Kramer, T.R., The Off-Line Programming System (OLPS): A Prototype STEP-Based NC-Program Generator, proceedings of a seminar *Product Data Exchange for the 1990s*, New Orleans, Louisiana, NCGA, February 1991, Vol. 2

[Kramer91b] Kramer, T.R., An Express Schema for Machining Plugged into ALPS4, unpublished, January 1991

[Kramer92] Kramer, T.R., A Library of Material Removal Shape Element Volumes (MRSEVs), NISTIR 4809, National Institute of Standards and Technology, March 1992

[Li90} Li, R-K., and Yu, M-H., A Framework for Prismatic Part Data Generation - Unit-Machined Loop Concept, *International Journal of Computer Integrated Manufacturing*, Vol. 3, No. 2, pp. 96-111

[Mortenson85] Mortenson, M. E., Geometric Modeling, John Wiley & Sons, New York, 1985

[Radack91] Radack, G., and Winn, A., Feature-Based Design Research, proceedings of a *Workshop on Feature-Based Design and Manufacturing*, sponsored by Martin Marietta Energy Systems, held in Gatlinburg, Tennessee, June 1991

[Ray91] Ray, S. and Catron, B., ALPS - A Language for Process Specification, *International Journal of Computer Integrated Manufacturing*, Vol 4, No. 2, pp 105-113

[Spiby91] Spiby, P., EXPRESS Language Reference Manual, ISO TC184/SC4 Document N14, April 1991